

THE MARKOFF-AUTOMATON

A NEW ALGORITHM FOR SIMULATING THE TIME-EVOLUTION OF LARGE STOCHASTIC DYNAMIC SYSTEMS

THOMAS FRICKE*

Institut für Physik

Humboldt-Universität

Invalidenstr. 110, D-100 99 Berlin, Germany

and

DIETMAR WENDT†

Institut für Theoretische Physik

Rheinisch Westfälische Technische Hochschule Aachen

Sommerfeldstraße, D-52 056 Aachen, Germany

Abstract

We describe a new algorithm for simulating complex Markoff-processes. We have used a reaction-cell method in order to simulate arbitrary reactions. It can be used for any kind of RDS on arbitrary topologies, including fractal dimensions or configurations not being related to any spatial geometry. The events within a single cell are managed by an event handler which has been implemented independently of the system studied. The method is exact on the Markoff level including the correct treatment of finite numbers of molecules.

To demonstrate its properties, we apply it on a very simple reaction-diffusion-systems (RDS). The chemical equations $A+A\rightarrow\emptyset$ and $A+B\rightarrow\emptyset$ in 1 to 4 dimensions serve as models for systems whose dynamics on an intermediate time scale are governed by fluctuations. We compare our results to the analytic approach by the scaling ansatz. The simulations confirm the exponents of the $A+B$ system within statistical errors, including the logarithmic corrections in the dimension $d=2$.

The method is capable to simulate the crossover from the reaction to diffusion limited regime, which is defined by a crossover time depending on the system size.

1 Introduction

Thus the dynamics of chemical reactions in homogeneous systems is well understood, they are easy to describe by a van't Hoff ansatz [1]. However, in inhomogeneous systems the transport

*Email thomas@summa.physik.hu-berlin.de

†Email dietmar@acds16.physik.rwth-aachen.de

mechanism has to be taken into account. We want to consider reaction–diffusion–systems (RDS) for which the slowest mechanism determines the dynamics of a RDS. RDS are known to be capable of a very complex behaviour like formation of spatial and temporal patterns. They show this great variety of effects, even if their discrete nature can be neglected and they are well described by the mean concentrations $n_\alpha(\vec{r})$. Population dynamics, which is often described by a formulation equivalent to chemical reactions, deals with much smaller concentrations, so the role of fluctuations is much more important than for chemical reactions.

1.1 The structure of this paper

At first, to introduce the notation, we briefly review the general method for simulating master–equations, with respect to chemical reactions. Then we will present the reaction–cell model to describe extensive reaction volumes. After that, we give a detailed description of the implementation as a the "Markoff–automaton". Our algorithm makes use of non–numerical data structures borrowed from computer science, because we need programming methods not commonly used in standard numerics. Therefore, we describe our methods in details. They may be regarded as "semi–numerical" algorithms in the sense of [2]. Finally, we will present the results we have gained applying our algorithm to the simplest RDS $A + A \rightarrow \emptyset$ and $A + B \rightarrow \emptyset$ in 1 to 4 dimensions.

1.2 Reaction–Diffusion–Systems

We want to denote different types of molecules Ξ_α by a Greek index α , while the reactions are distinguished by the Latin letter i . The stoichiometric equations describe the chemical reaction of the i –th type unambiguously by the initial (or forward) f_α^i and final (or backward) b_α^i coefficients and the related reaction–rate λ^i

$$\sum_{\alpha} f_{\alpha}^i \Xi_{\alpha} \xrightarrow{\lambda^i} \sum_{\alpha} b_{\alpha}^i \Xi_{\alpha}. \quad (1)$$

The common formulation of the mean–field equations for position–dependent mesoscopic concentrations $n(x, t)$ leads to nonlinear coupled partial differential equations

$$\frac{\partial}{\partial t} n_{\alpha} = D_{\alpha} \Delta n_{\alpha} + \sum_i \lambda^i \prod_{\beta} (b_{\beta}^i - f_{\beta}^i) n_{\beta}^{f_{\beta}^i}. \quad (2)$$

In some realisations of certain systems they may also describe the fluctuations [3, 4] or give only quantitative changes [5], nevertheless, it may become necessary to respect the influence of fluctuations, especially in biological systems, where the number of individual molecules is very small [6, 7].

However, in contrast to the simple derivations of the partial differential equations for the mean values, the formulation of the equations for the higher moments are much more complicated, even for simple systems.

Note the different scaling of mean concentrations and fluctuations. While concentrations are proportional to the number of particles N_{α} in a volume Ω , i.e $N_{\alpha} = \Omega n_{\alpha}$, fluctuations scale with a different power law according to $\delta N_{\alpha} \cong \sqrt{N_{\alpha}}$. If we introduce the fluctuations of n_{α} as additional quantities δn_{α} , we have to take into account, that they usually scale as $\delta n_{\alpha} \cong \sqrt{N_{\alpha}}$. However, the assumption of a $\Omega^{1/2}$ scaling is invalid, if fluctuations may increase, as in our case. Other important examples of scaling departing from this law are physical systems showing critical behaviour. A rigorous derivation of the fluctuation scaling by terms of an $1/\Omega$ expansion is given in [1].

Therefore, the analytical treatment turns out to become very difficult, if diffusion and the discrete nature of the molecules together must be taken into account. Not only the cost of computation grows proportional to the volume, it has to be remarked that it also increases exponentially to the statistical moments taken into consideration.

In the reaction-limited case, diffusion is sufficiently fast to keep the concentrations homogeneous. Thus the reactions follow a global van't Hoff dynamics, for which diffusion may be neglected. For the diffusion-controlled reactions the time to pass structures like domains or empty spaces determines the time scale of the RDS. To introduce spatial resolution we subdivide the volume Ω into L^d small cells of size $\omega = \Omega/L^d$.

Therefore, for small numbers of molecules within an ω volume, the discrete nature of the molecules must be taken into consideration. The raw estimation for systems of N molecules leads to an estimation for the fluctuations of $O(\sqrt{N})$. Thus we need $O(N) = 10,000$ molecules to guarantee a relative error of percentage order if fluctuations are neglected.

Former simulations [8, 9, 10] using cellular automata or lattice gas methods to study the dynamics of RDS are restricted to small volumes and small numbers of molecules. Computer simulations of complex RDS focus on the mean-field-behaviour taking no notice of the fluctuations, whereas on the other hand stochastical simulations of extensive systems close to equilibrium are not well suited to dynamics. In contrast, using our algorithm we may handle some 10^7 particles and are, furthermore, able to study the crossover from the diffusion-controlled to the reaction-controlled limit. Following an analysis of the relevant time scales, the diffusion can be simulated on a coarse lattice of cells without affecting the exactness of the results. Our goal is to close the gap between solving the mean-field-equations and simulation techniques using cellular automata. Both techniques are included, but we want to emphasize, that standard algorithms may be more efficient to cover these limits. Because our method is related to cellular automata, we call it a "Markoff-automaton" claiming its exactness on the level of the Markoff-processes derived.

The algorithm has been written for systems with a clear separation of the short time-scale of reaction events and the long time-scale of the decay of the population. It should be used with care, when the scales are mixed, f.e. by local ordering phenomena.

1.3 The physical systems

We consider two different RDS consisting of one of the two simplest chemical reactions:



1.4 The reaction-limited case

In this case, the diffusion is as fast, that the extinction is controlled by the reaction. Thus both systems may be described by a global van't Hoff ansatz, which leads to the differential equations for the concentrations n_α

$$\frac{d}{dt}n_A(t) = -k n_A^2(t), \quad (5)$$

$$\frac{d}{dt}n_{A,B}(t) = -k n_A(t)n_B(t). \quad (6)$$

The straightforward solution of (5) reads

$$n_A(t) = \frac{n_A(0)}{n(0)kt + 1}. \quad (7)$$

To solve (6), we take advantage of the fact, that the difference $\Delta n(t) = n_A(t) - n_B(t) = \Delta n(0)$ is conserved. This way we obtain the mean-field result

$$n_A(t) = \frac{n_A(0)\Delta n \exp(-\Delta nkt)}{n_B(0) - n_A(0) \exp(-\Delta nkt)}, \quad (8)$$

$$n_B(t) = n_A(0) + \Delta n. \quad (9)$$

For the case of equal initial concentrations, i.e $\Delta n = 0$, equation (8) reduces to (7). Note that the initial condition $n_A(0) = n_B(0)$ is responsible for the algebraic decay.

1.4.1 The diffusion limited case for $A + A \rightarrow 0$

In this case the extinction is controlled by the speed of diffusion, which has been analysed in detail by the scaling ansatz of Kang and Redner [9, 10]. However, at this point, we can only give a brief description of their main arguments, because we have outlined their scaling in the appendixes.

While we want to use this simple model to test our algorithm, we also want to examine the accuracy of the scaling ansatz, which predicts an algebraic annihilation for the diffusion controlled $A + A$ reaction. The time t_ξ is the time to pass the mean distance between two A molecules at $t = 0$,

$$n_A(t) \propto \begin{cases} t^{-d/2}, & d < 2, \\ t^{-1}, & d \geq 2. \end{cases} \quad t > t_\xi, \quad (10)$$

This RDS shows a critical dimension $d_c = 2$, for $d \geq d_c$ the decay is predicted to be determined by mean field behaviour.

1.4.2 The diffusion limited case for $A + B \rightarrow 0$

In the $A + B$ reaction we obtain the time scale t_ξ determined by the time to pass a domain as

$$t_\xi = \frac{1}{D} \left| \sqrt{n_A(0)} - \sqrt{n_B(0)} \right|^{-2/d}. \quad (11)$$

An algebraic annihilation $n_{AB} \propto t^{-\frac{1}{4}}$ with $d_c = 4$ appears on a time scale $t < t_\xi$. Thus, if $n_A(0) < n_B(0)$ we obtain in the limit $\Delta n = n_A(0) - n_B(0) \rightarrow 0$,

$$n_{A,B}(t) \propto \begin{cases} t^{-d/4}, & d < 4, \\ t^{-1}, & d \geq 4. \end{cases} \quad t < t_\xi, \quad (12)$$

A more detailed explanation of the scaling arguments is left to the appendixes. Both of these simple reactions are the most important examples for the influence of fluctuations in RDS. Therefore, to improve our understanding of RDS beyond mean-field-behaviour, we present our simulation method as an approach correctly treating dynamics and statistics in a natural way.

2 Simulating master-equations

We want to treat a chemical reaction as a Markoff-process, which is described by the time evolution of the probability $p_{\vec{X}}$ to be in state \vec{X} at time t following a stationary master-equation

$$\frac{dp_{\vec{X}}}{dt} = \sum_{\vec{X}'} \underbrace{\{W_{\vec{X}' \rightarrow \vec{X}} p_{\vec{X}'}\}}_{\text{inflow into } \vec{X}} - \underbrace{\{W_{\vec{X} \rightarrow \vec{X}'} p_{\vec{X}}\}}_{\text{outflow from } \vec{X}}, \quad (13)$$

in our case the discrete vector \vec{X} denoting the number of molecules [12].

The minimal-process-method simulates the master-equation as a random-walk in the space of all possible configurations. Starting the random-walk in state \vec{X} , we only need to determine the time leaving \vec{X} , and the successor state \vec{X}' , which is equivalent to draw the lifetime $\tau_{\vec{X}}$ and the transition $\vec{X} \rightarrow \vec{X}'$ as random numbers.

For both steps the knowledge of the flux out of \vec{X} is completely sufficient. The lifetime $\tau_{\vec{X}}$ is exponentially distributed, while the selection of the transition requires the drawing of a random-number \vec{X}' proportional to the transition rate $W_{\vec{X} \rightarrow \vec{X}'}$. This property is due to discrete Markoff-processes, and we want to emphasize that it is *exact* and not affected by any further assumptions.

To explain the name "minimal-process-method" it has to be remarked, that the master-equation (13) may be interpreted in different ways. We may look upon the outflow term as the death-process and upon the flux into a state as the birth-process of the state \vec{X} [14, 15]. The *minimal-process* being the process with the least number of events is the random-walk-process in the discrete configuration space of all possible \vec{X} . The minimal process distinguishes on the rule, that a death process in \vec{X} always coincides to a simultaneous birth process in \vec{X}' . With the sum

$$W_{\vec{X}} = \sum_{\vec{X}'} W_{\vec{X} \rightarrow \vec{X}'}, \quad (14)$$

the way to generate this random-walk and the sequence of states $\vec{X}(t)$ is given by the following simple algorithm:

Minimal-process-method

1. Generate an exponentially distributed random number τ with
 $\Pr(\tau) = W_{\vec{X}} \cdot \exp(-W_{\vec{X}} \cdot \tau), \quad \text{let } t \leftarrow t + \tau,$
2. select \vec{X}' according to the probability
 $\Pr(\vec{X} \rightarrow \vec{X}') = W_{\vec{X} \rightarrow \vec{X}'} / W_{\vec{X}},$
3. go back to step 1.

The characteristics of discrete processes were already known to Markoff, the minimal-process-method at least since the early sixties [14, 15]. However, no one has made use of it for computer simulations until the mid seventies when Gillespie applied it to chemical reaction-systems [12, 13].

2.1 Chemical reactions in a homogeneous volume

For the present, we want to discuss arbitrary chemical reactions in a small volume ω , for which the diffusion is so fast that any spatial inhomogeneity may be neglected. After that, we will introduce diffusion as a reaction-like process of molecules leaving the volume ω . In this way, we want to describe the dynamics by which diffusion relates distances of an extensive length scale to a time scale.

According to (1) the number X_α of molecules Ξ_α changes from X_α to $X_\alpha + b_\alpha^i - f_\alpha^i$ each time the reaction of the i -th type occurs. I.e. f_α^i molecules Ξ_α are consumed and b_α^i molecules Ξ_α are produced. Forward and backward reactions shall be distinguished by their enumerations. If the reactions j and i are reciprocals of each other, the role of the coefficients is exchanged, i. e. $f_\alpha^i = b_\alpha^j$, and $f_\alpha^j = b_\alpha^i$.

Since the number of molecules is an extensive quantity, the number of chemical reactions per unit time must be extensive, too. The rate of events, the reactivity Λ^i of a chemical

reaction is defined as

$$\Lambda^i dt = \Pr \left\{ \begin{array}{l} \text{the reaction } i \text{ occurs during} \\ \text{the time interval } [t, t + dt] \end{array} \right\}. \quad (15)$$

According to van't Hoff [1], it is given by

$$\Lambda^i = \lambda^i \omega \prod_{\alpha} \frac{X_{\alpha}!}{(X_{\alpha} - f_{\alpha}^i)! \omega^{f_{\alpha}^i}}, \quad (16)$$

where λ^i is an intensive constant which does not depend on X_{α} . Note that the reactivities Λ depend on the state, i.e. $\Lambda^i = \Lambda_{\vec{X}}^i$, $\Lambda = \Lambda_{\vec{X}}$, the index \vec{X} is usually suppressed. The fraction

$$\frac{X_{\alpha}!}{(X_{\alpha} - f_{\alpha}^i)! \omega^{f_{\alpha}^i}} = \frac{X_{\alpha}(X_{\alpha} - 1) \cdots (X_{\alpha} - f_{\alpha}^i + 1)}{\omega^{f_{\alpha}^i}} \quad (17)$$

takes into consideration that each individual molecule can only be consumed once. This is important, if the numbers X_{α} are small. For large X_{α} , this fraction turns into the more familiar form $\left(\frac{X_{\alpha}}{\omega}\right)^{f_{\alpha}^i}$, ignoring any power up to $\left(\frac{X_{\alpha}}{\omega}\right)^{f_{\alpha}^i-1}$.

The van't Hoff approach suggests chemical reactions being Markoff-processes [12, 13], which may be justified due to a stosszahlansatz for homogeneous concentrations. The configuration is the vector consisting of the number of molecules

$$\vec{X} = (X_1, X_2, \dots, X_{\alpha}, \dots). \quad (18)$$

For abbreviation we group the initial and final coefficients for the reaction of the i -th type as vectors $\vec{f}^i = (f_1^i, f_2^i, \dots, f_{\alpha}^i, \dots)$, $\vec{b}^i = (b_1^i, b_2^i, \dots, b_{\alpha}^i, \dots)$. Each type of reaction is realized by an integer vector operation, the vector $\vec{\delta}^i = \vec{b}^i - \vec{f}^i$ denoting the changes caused by the i -th reaction:

$$\vec{X} \leftarrow \vec{X} + \vec{\delta}^i. \quad (19)$$

In our case the vector \vec{X} is simply a one- or two-dimensional vector

$$\vec{X} = \begin{cases} (X_A) & \text{reaction } A + A \rightarrow \emptyset, \\ (X_A, X_B) & \text{reaction } A + B \rightarrow \emptyset. \end{cases} \quad (20)$$

The reaction-rates are computed according (16) as

$$\Lambda_{AA} = \lambda_{AA} \omega \frac{X_A(X_A - 1)}{\omega^2}, \quad (21)$$

$$\Lambda_{AB} = \lambda_{AB} \omega \frac{X_A X_B}{\omega^2}. \quad (22)$$

The homogeneity of the concentrations within ω is necessary for the van't Hoff ansatz. We will have to subdivide the volume into sufficiently small cells to justify a local form of (16), introducing diffusion as hopping from one cell to an adjacent one.

2.2 Gillespie's algorithm

With these assumptions Gillespie describes the sequence of states of a chemical system by the random-walk of the vector $\vec{X} = \vec{X}(t)$. His algorithm is a version of the minimal-process-method adapted to the dynamics of chemical reactions, which is based on the well known properties of discrete stochastic processes[14, 15, 12, 13]. We summarise his main results according to our requirements. The total reactivity is the rate

$$\Lambda = \sum_{\text{all reactions } i} \Lambda^i, \quad (23)$$

which determines the probability of a chemical reaction during the infinitesimal time interval $[t, t + dt]$ according to

$$\Lambda dt = \Pr \left\{ \begin{array}{l} \text{some chemical reaction changes the} \\ \text{state } \vec{X} \text{ during } [t, t + dt] \end{array} \right\}. \quad (24)$$

Λ is the inverse of the average lifetime of \vec{X} , therefore the random time τ_Λ until the next event occurring has the probability density

$$\Pr\{\tau_\Lambda, \text{time to next reaction}\} = \Lambda \exp(-\Lambda \tau_\Lambda). \quad (25)$$

Because we have assumed that chemical reactions are Markoff-processes, this equation yields independently of the time when the last change occurred, which is a general statement for any discrete Markoff-process. According to Gillespie, the probability of the reaction of kind i is proportional to its contribution to Λ , thus

$$\Pr\{\text{Reaction } i\} = \frac{\Lambda^i}{\Lambda} \quad (26)$$

is the probability, that the next change of \vec{X} in the chemical system is determined by a reaction of type i defined by (19). The transition rate $W_{\vec{X} \rightarrow \vec{X}'}$ is given by the probability of all chemical reactions leading from \vec{X} to \vec{X}'

$$W_{\vec{X} \rightarrow \vec{X}'} = \sum_{\substack{\text{all reactions } i \text{ with} \\ \vec{X}' = \vec{X} - \vec{f}^i + \vec{b}^i}} \Lambda^i. \quad (27)$$

Note that different chemical reactions may change \vec{X} by the same $\vec{\delta}^i = \vec{f}^i - \vec{b}^i$ because only the backward and forward stoichiometric coefficients \vec{f}^i and \vec{b}^i together determine the reaction unambiguously.

Gillespie's algorithm implicitly uses equation (27) to decompose the relation of reaction and transition probabilities. For arbitrary chemical reactions it computes the random time-step τ_Λ until the next reaction by drawing a uniformly distributed random number $\text{rnd} \in [0, 1)$. The reaction type i usually is selected by a simple loop, known as linear selection algorithm using the sum s for the integration. The variable t denotes the time, proceeding in exponential time steps. The order of reactions denoted by i is arbitrary. After the reaction has been carried out, the reaction reactivities Λ^i and the total rate $\Lambda = W_{\vec{X}}$ are computed again.

Gillespie's algorithm

1. $\tau_\Lambda \leftarrow -\frac{1}{\Lambda} \log(1 - \text{rnd})$, $t \leftarrow t + \tau_\Lambda$
2. $r \leftarrow \text{rnd}$, $s \leftarrow 0$, $i' \leftarrow \text{first reaction}$,
3. while $s \leq r\Lambda$,

(a)	$i \leftarrow i'$, $i' \leftarrow \text{next reaction}$
(b)	add up $s \leftarrow s + \Lambda^i$
4. do reaction type i by $\vec{X} \leftarrow \vec{X} + \vec{\delta}^i$, compute Λ^i , and $\Lambda \leftarrow \sum_i \Lambda^i$,
5. go back to step 1.

Note the adaptation to the time scale Λ^{-1} , which is responsible for the high flexibility of Gillespie's algorithm. Thus it is a fast and easy to use Monte-Carlo method for the dynamics of arbitrary reaction-systems and has the main advantage over the mean-field description, that it takes into account internal fluctuations caused by the finite numbers of molecules. Because the random-walk imitates the sequence of states of the simulated Markoff-process, any correlations and higher moments may be measured as in a real experiment.

3 The reaction–cell method

The molecules are treated as point–like particles with a finite interaction probability, implicitly assuming, that their Brownian motion is independent. Our simulation introduces a reaction–cell method. In general, for small diffusion–constants, the homogeneity condition of the van’t Hoff ansatz cannot be fulfilled for the total d –dimensional volume Ω . Thus Ω is subdivided into small cells (cubes or squares etc.) of volume $\omega = \frac{\Omega}{L^d} = h^d$. We have to impose the condition, that the size of ω may be justified by a local van’t Hoff ansatz. This prerequisite depends on the time–scale τ_R between two subsequent chemical reactions compared to the time scale τ_D of a molecule leaving ω by diffusion. The diffusion–time–scale has to be much faster than the reaction time–scale, i. e. $\tau_D \ll \tau_R$ with $\tau_D = h^2/D$ and $\tau_R = 1/\Lambda$.

Our algorithm covers the field between the deterministic description of RDS, which does not describe the fluctuations correctly, and the cellular automaton approach, which is not well conditioned to describe large numbers of particles. The assumptions have to be justified in any case, and have to be modified carefully, f.e. if the spatial extension of the moving objects has to be taken into account. This is important for the $A + A \rightarrow 0$ reaction in lower dimensions [11, 16], where the seed reaction constant may become a irrelevant quantity. In this case first principle Monte–Carlo simulations have to be performed and our algorithm has to be checked.

3.1 Reaction–Diffusion–Systems

We distinguish different cells by an integer vector index \vec{n} . Therefore, the possible reactions now depends on the local situation which is described by the number $X_{\alpha\vec{n}}$ of molecules Ξ_α in cell \vec{n} . The total volume Ω is represented by the direct sum

$$\tilde{X} = \bigoplus_{\vec{n}} \vec{X}_{\vec{n}} = \bigoplus_{\vec{n}} \bigoplus_{\alpha} X_{\alpha\vec{n}}, \quad (28)$$

consisting of sub–vectors $\vec{X}_{\vec{n}} = \{\dots, X_{\alpha\vec{n}}, \dots\}$ of the numbers of molecules in each cell.

For the $A + A \rightarrow \emptyset$ system \tilde{X} has the form

$$\tilde{X} = \left\{ (X_{A(1,\dots,1)}), (X_{A(1,\dots,2)}), \dots, (X_{A(L,\dots,L)}) \right\}, \quad (29)$$

respectively for the $A + B \rightarrow \emptyset$ reaction

$$\begin{aligned} \tilde{X} = \left\{ (X_{A(1,\dots,1)}, X_{B(1,\dots,1)}), (X_{A(1,\dots,2)}, X_{B(1,\dots,2)}), \right. \\ \left. \dots, (X_{A(L,\dots,L)}, X_{B(L,\dots,L)}) \right\} \end{aligned} \quad (30)$$

In this section we want to introduce diffusion as a random–walk on the lattice of reaction–cells. The diffusion in an arbitrary, not necessarily euclidian, topology may be regarded as a reaction–like step of a molecule Ξ_α from \vec{n} to one of its next neighbours \vec{m} ,

$$\Xi_{\alpha\vec{n}} \xrightarrow{\lambda_{\alpha\vec{n}\vec{m}}} \Xi_{\alpha\vec{m}}. \quad (31)$$

For a diffusion–step of a molecule Ξ_α the rate

$$\lambda_{\alpha\vec{n}\vec{m}} = D_{\alpha\vec{n}\vec{m}} \frac{1}{h^2}, \quad (32)$$

with $D_{\alpha\vec{n}\vec{m}}$ denoting the local diffusion–constant, describes the probability

$$\lambda_{\alpha\vec{n}\vec{m}} dt = \Pr \left\{ \begin{array}{l} \text{a single molecule } \Xi_\alpha \text{ jumps from } \vec{n} \text{ to } \\ \vec{m} \text{ during the time intervals } [t, t + dt] \end{array} \right\}. \quad (33)$$

On a lattice built of volumes $\omega = h^d$, we obtain in the limit $t \rightarrow \infty$ the correct Brownian motion behaviour by the central-limit-theorem. The symbol $\langle \vec{n} \rangle$ denotes the set of all $2d$ next neighbour cells of \vec{n} . Due to the symmetry of the rate to do a single step into any direction i.e. the rate of leaving \vec{n} by diffusion, is $D_{\alpha\vec{n}}/h^2$, with

$$\lambda_{\alpha\vec{n}} = \sum_{\vec{m} \in \langle \vec{n} \rangle} \lambda_{\alpha\vec{n}\vec{m}} = 2d\lambda_{\alpha\vec{n}\vec{m}}. \quad (34)$$

In the isotropic and homogeneous case we omit the spatial indices $\lambda_\alpha = \lambda_{\alpha\vec{n}}$ and the rate for stepping in a certain dimension gets $\lambda_\alpha/2d$, while the total rate for leaving a cell is λ_α .

If we have $X_{\alpha,\vec{n}}$ molecules Ξ_α in cell \vec{n} , the rate for the next diffusion step of any Ξ_α to \vec{m} is

$$\Lambda_{\vec{n}\vec{m}} = \lambda_{\vec{n}\vec{m}} X_{\alpha\vec{n}}. \quad (35)$$

This rate is the equivalent formulation to (16), treating a diffusion-step like a first order chemical reaction. If a diffusion-step occurs, another cell \vec{m} gets involved, because one Ξ_α moves from \vec{n} to \vec{m} . Thus a diffusion-step is carried out by

$$1. X_{\alpha\vec{n}} \leftarrow X_{\alpha\vec{n}} - 1, \quad 2. X_{\alpha\vec{m}} \leftarrow X_{\alpha\vec{m}} + 1. \quad (36)$$

Notice that in our case $\Lambda_{\alpha\vec{n}\vec{m}}$ depends only on the contents of \vec{n} , from which a molecule hops into a neighbour $\vec{m} \in \langle \vec{n} \rangle$, although the assigned diffusion-step changes the contents of the other cell \vec{m} , too. Thus a diffusion-step changes the state of two cells, however, its rate depends only on the contents of the first cell \vec{n} triggering the event.

For the chemical reaction of type i we simply have to change the contents of \vec{n} by $\vec{X}_{\vec{n}} \leftarrow \vec{X}_{\vec{n}} + \vec{\delta}_{\vec{n}}^i$. Therefore, the probability $Q_{\vec{n}}$ of an event triggered by cell \vec{n} is the sum of reaction-rates $\Lambda_{\vec{n}}^i$ and diffusion-rates $\Lambda_{\alpha\vec{n}\vec{m}}$

$$Q_{\vec{n}} = \sum_i \Lambda_{\vec{n}}^i + \sum_\alpha \sum_{\vec{m} \in \langle \vec{n} \rangle} \Lambda_{\alpha\vec{n}\vec{m}} \quad (37)$$

Because all reaction- and diffusion-steps are assigned to the cell \vec{n} , we may identify an arbitrary step entirely by an integer i and the cell \vec{n} . We may ignore the additional index \vec{m} for the selection mechanism. A reaction may be defined by an addition of its associated difference $\vec{\delta}_{\vec{n}}^i$, i.e. $\vec{X} \leftarrow \vec{X} + \vec{\delta}_{\vec{n}}^i$, involving one or two cells. In section 6.1.1 below we suggest obvious generalisations of rates depending on the contents of the neighbourhood of \vec{n} . This way it is easy to introduce dynamics depending on gradients, like the movement of kinks and steps on surfaces.

3.2 Times scales of reaction- and diffusion-processes

If we want to fulfill the assumptions of a local van't Hoff ansatz we have to satisfy the condition, that diffusion is much faster than any chemical reaction. More precisely spoken, this means, that the local reactions $\Lambda_{\vec{n}}^i$ and diffusion- $\Lambda_{\alpha\vec{n}\vec{m}}$ rates define different time scales, which have to be separated. Therefore we choose the size of our cells sufficiently small, for the condition

$$\Lambda_{\vec{n}}^i \ll \Lambda_{\alpha\vec{n}\vec{m}}, \quad \text{for any } i, \alpha \quad (38)$$

to be satisfied. This can always be achieved, because reaction probabilities are extensive quantities $\Lambda_{\vec{n}}^i \propto \omega = h^d$, while diffusion-rates increase with h^{-2} . Thus $\lambda_{\alpha\vec{n}\vec{m}} \propto \omega^{-2/d}$ and $\Lambda_{\alpha\vec{n}\vec{m}} \propto \omega^{1-2/d}$, the lattice constant h always can be reduced to make the ratio

$$\frac{\Lambda_{\vec{n}}^i}{\Lambda_{\alpha\vec{n}\vec{m}}} \propto \omega^{2/d} = h^2 \quad (39)$$

arbitrarily small. For any practical purpose it is not possible to give an a priori estimation, so we recommend to choose ω as large as possible rejecting the simulation runs if $10\Lambda_{\vec{n}}^i > \Lambda_{\vec{n}\vec{m}}^j$.

4 Selecting a single cell by the method of logarithmic classes

Although we have argued that there is no principal difference between the Gillespie algorithm in a homogeneous volume and the method on a lattice of reaction cells, a significant complication arises as the number of possible changes in each Markoff-step is an extensive quantity. If we make a straightforward generalisation of Gillespie's algorithm using a linear selection strategy we run into the problem of selecting a single event from an extensive quantity of transitions. The total reactivity

$$Q = \sum_{\vec{n}} Q_{\vec{n}} = \sum_{\vec{n}} \sum_i Q_{\vec{n}}^i \quad (40)$$

again determines the mean time step. According to condition (38) we choose the sequence of transitions considering at first the more probable diffusion-steps, afterwards the possibility of chemical reactions.

Gillespie's algorithm for reaction-cells with linear selection

1. $\tau_Q \leftarrow -\frac{1}{Q} \log(1 - \text{rnd})$, $t \leftarrow t + \tau_Q$,
2. $r \leftarrow \text{rnd}$, $s \leftarrow 0$,
3. while $s < rQ$, for all cells \vec{n} ,

- (a) next \vec{n} ,
 - (b) while $s < rQ$,

- i. next i ,
 - ii. sum $s \leftarrow s + Q_{\vec{n}}^i$,
4. do transition $\tilde{X} \leftarrow \tilde{X} + \tilde{\delta}_{\vec{n}}^i$,
5. for all cells \vec{n} and reactions i involved,
recompute all $Q_{\vec{n}}^i \Rightarrow Q_{\vec{n}} \Rightarrow Q$,
6. go back to step 1.

This method suffers from the severe disadvantage that the sum S computed at step (3) adds up an extensive number of reaction probabilities. Drawing a random cell due to its contribution $Q_{\vec{n}}$ to Q , the loop consumes computing time proportional to the number of cells. This procedure is completely unsuitable for a single step of a computer algorithm, see figure 1(a).

If we do not have any additional information, we have on the average to walk the half length of the main loop. Even for a small lattice of $10^4 = 100 \times 100$ cells this means an increase for the computing time compared to the algorithm without spatial resolution by an average factor of 5,000. It seems to be impossible to circumvent this problem by rearranging the cells or some kind of pre-sorting, because the selected reaction changes the cell-reactivity every time. A random selection according A. J. Walker's [2] algorithm would be efficient only if the reaction-rates did not depend on time. The extensive quantity of cells to be considered is a principle obstacle.

These reflections show that linear selection is totally inefficient. Thus we have tried another algorithm based on the von Neumann rejection, which selects at first the reaction-cell and in a second step the reaction or diffusion in this cell.

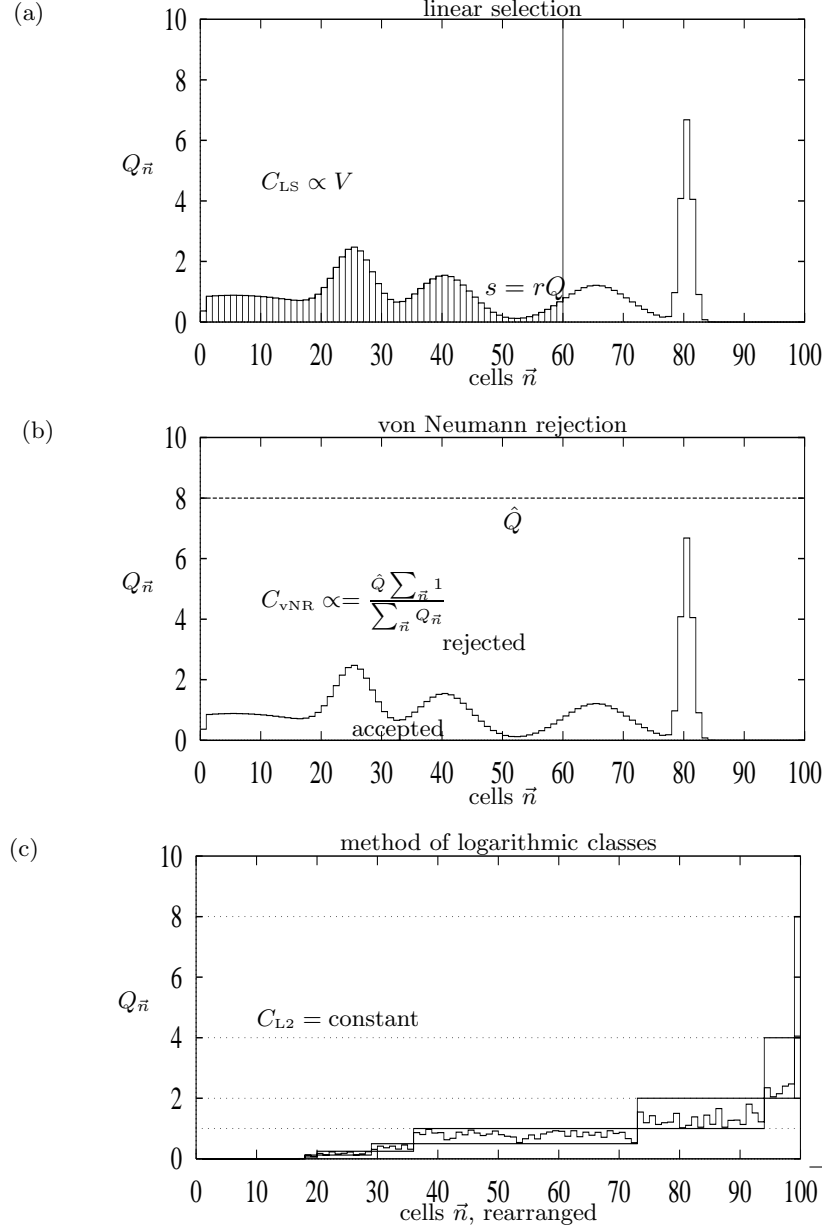


Figure 1: Selection algorithms: (a) Linear s., the loop integrates $Q_{\vec{n}}$ up to rQ , thus the cost is extensive, (b) Von Neumann rejection without improvement, the cost is the ratio of the area of the rectangle marked off by the upper limit \hat{Q} to the area below $Q_{\vec{n}}$. A single peak may severely affect its efficiency, (c) s. by the method of logarithmic classes, the upper and lower limits for each class are denoted by dotted lines. The acceptance $a \geq 0.5$ is the ratio of the reordered $Q_{\vec{n}}$ to its upper estimation $2^{\text{floor}(\text{ld}(Q_{\vec{n}})) + 1}$, we find the cost being approximately constant.

4.1 Von Neumann rejection

The von Neumann rejection requires an upper bound for the probability to select a cell. Therefore we have to make the assumption, that

$$Q_{\vec{n}} \leq \hat{Q} \quad \text{for each cell } \vec{n}. \quad (41)$$

We replace step (3) by

(3) von Neumann rejection

(a) select a cell \vec{n} with uniform probability,
(b) choose a uniformly distributed random number $r \leftarrow \text{rnd}$,
(c) if $Q_{\vec{n}} \geq r\hat{Q}$ then select a reaction according to $X_{\alpha\vec{n}}$
else reject this choice and go back to step (a).

On the first glance this algorithm seems to solve all problems. The method does not depend on the number of cells, thus the problem of the dependency on the extensive number of cells does not arise. However, a more precise look reveals that the efficiency of the von Neumann rejection strongly depends on the homogeneity of the RDS. This is demonstrated in figure 1(b). The acceptance ratio a is the quotient of the area below $Q_{\vec{n}}$ and the rectangle delimited by \hat{Q}

$$a = \frac{\sum_{\vec{n}} Q_{\vec{n}}}{\hat{Q} \sum_{\vec{n}} 1}, \quad (42)$$

and $1/a$ is proportional to the number of runs through the loop needed to select a cell. Therefore the cost C_{vnr} of the algorithm is proportional to the inverse of the acceptance ratio,

$$C_{\text{vnr}} \propto \frac{1}{a} = \frac{\hat{Q} \sum_{\vec{n}} 1}{\sum_{\vec{n}} Q_{\vec{n}}}. \quad (43)$$

For very inhomogeneous systems, i.e. for systems dominated by a single peak, this method may slow down by an arbitrary factor, which has been studied for the simulation of a biological system [6]. This disaster can be avoided, if we split up the cells into classes according to their dual order of magnitude. The acceptance ratio is improved to 75%.

4.2 Method of logarithmic classes

If we want to handle all orders of magnitude of the cell reactivities $Q_{\vec{n}}$ we have to implement a logarithmic classification scheme. We define the logarithmic class L_z as the set of all cells \vec{n} with a reactivity $Q_{\vec{n}}$ in the same order of dual magnitude. The symbol $\text{ld}(x) = \log_2(x)$ denotes the logarithm with base 2, and the function $\text{floor}(x)$ denotes the largest integer which is not greater than x

$$L_z = \{\vec{n} \mid \text{floor}(\text{ld}(Q_{\vec{n}})) = z\}. \quad (44)$$

The reactivity ℓ_z of a certain class is given by the sum of the reactivities of its elements

$$\ell_z = \sum_{\vec{n} \in L_z} Q_{\vec{n}}. \quad (45)$$

There is no principal restriction of the range of z

$$z = \text{floor}(\text{ld}(Q_{\vec{n}})) \in \{-\infty, \dots, -2, -1, 0, 1, 2, 3, \dots\}. \quad (46)$$

The class $L_{-\infty}$ contains all cells without any possibility of a reaction– or diffusion–step, above all the empty cells. Because $L_{-\infty}$ cannot be selected, i.e. $\ell_{-\infty} = 0$, there is no need to represent it in computer memory. For any practical application, z is restricted to a finite range $z_{\min} \leq z \leq z_{\max}$. The intersection of two different classes is empty $L_z \cap L_{z'} = \emptyset$, $z \neq z'$, thus the total reactivity is expressed by the sum

$$Q = \sum_{\vec{n}} Q_{\vec{n}} = \sum_{z=-\infty}^{\infty} \ell_z = \sum_{z=z_{\min}}^{z_{\max}} \ell_z. \quad (47)$$

The inequality

$$2^z \leq Q_{\vec{n}} < 2^{z+1}, \quad \vec{n} \in L_z \quad (48)$$

guarantees, that the reaction–rates are relatively homogeneous within a certain class. The rates of two members of any class do not differ by more than a factor of 2

$$\frac{\max_{\vec{n} \in L_z} Q_{\vec{n}}}{\min_{\vec{n}' \in L_z} Q_{\vec{n}'}} < 2. \quad (49)$$

This is an ideal starting point for the von Neumann rejection. We therefore decided to implement an event handler, which is able to select a reaction–cell according to its reaction–rate by a von Neumann rejection step within the class. The number of classes is small, i.e. of $O(10)$, thus the choice of a class is based on a linear selection algorithm.

4.2.1 Selecting a reaction

The problem has been split into three qualitatively different steps:

1. select a class L_z with probability ℓ_z/Q ,
2. select a cell $\vec{n} \in L_z$ with probability $Q_{\vec{n}}/\ell_z$,
3. select a reaction–step i within cell \vec{n} with probability $Q_{\vec{n}}^i/Q_{\vec{n}}$.

We stress again, that the index i represents both reaction and diffusion transitions. The last two steps are selected according their conditional probabilities

$$\Pr\{\text{select class } z\} = \frac{\ell_z}{Q}, \quad (50)$$

$$\Pr\{\text{select cell } \vec{n} \mid \text{class } z \text{ has been selected}\} = \frac{Q_{\vec{n}}}{\ell_z}, \quad (51)$$

$$\Pr\{\text{select reaction } i \mid \text{cell } \vec{n} \text{ has been selected}\} = \frac{Q_{\vec{n}}^i}{Q_{\vec{n}}}, \quad (52)$$

therefore the probability of selecting a reaction in a cell

$$\Pr\{\text{select reaction } i \text{ in cell } \vec{n}\} = \frac{\ell_z}{Q} \frac{Q_{\vec{n}}}{\ell_z} \frac{Q_{\vec{n}}^i}{Q_{\vec{n}}} = \frac{Q_{\vec{n}}^i}{Q} \quad (53)$$

has been maintained correctly.

The algorithm choosing a class is a linear selection. However, it is not necessary to order the sequence of classes (z_i, \dots, z_j) in a naive way with $z_i < z_{i+1}$ or $z_i > z_{i+1}$ for all i . Moreover, it has a favourable effect, if the classes are sorted with respect to their reactivity, i.e. $\ell_{z_i} \geq \ell_{z_{i+1}}$. To speed up the selection, the classes of the highest probabilities to be selected are successively moved to the head of the loop by a bubble–sort, see step (2c) below. This is the most efficient algorithm to select a reaction in a class according to its rate ℓ_{z_i} , because the classes with the smallest probabilities to be drawn least are moved to the tail.

1. Selection of a logarithmic class L_z

1. random number $r \leftarrow \text{rnd}$, sum $s \leftarrow 0$, index $i = 0$
2. while $s \leq rQ$
 - (a) increase $i \leftarrow i + 1$,
 - (b) add up $s \leftarrow s + \ell_{z_i}$,
 - (c) if $i > 1$ and $\ell_{z_i} > \ell_{z_{i-1}}$ then
exchange the order of the classes z_i and z_{i-1} .
3. return $z = z_i$, i.e draw L_{z_i} .

After z has been selected, it is easy to propose a cell \vec{n} by drawing a uniform random number $u \in \{0, \dots, \nu_z - 1\}$. We have implemented each class L_z as an array $F_z[0, \dots, \nu_z - 1]$ of $\nu_z = |L_z|$ elements, each describing the state of a single cell. The following subroutine does the von Neumann rejection of a cell in the previous selected class z .

2. Selecting a cell \vec{n} in class L_z by von Neumann rejection

1. propose $u \leftarrow \text{floor}(\nu_z \text{rnd})$, $\vec{n} \leftarrow F_z[u]$,
2. draw a uniform random number $r \leftarrow \text{rnd}$,
3. if $Q_{\vec{n}} > r2^{z+1}$ then return \vec{n} ,
else go back to step 1.

Because of the inequality (48) we know lower and upper limits of the reactivity, $2^z \leq Q_{\vec{n}} < 2^{z+1}$. Therefore, it is guaranteed, that the probability for a rejection is less than 0.5. If we assume that the rates in each class are distributed uniformly, we get an acceptance ratio $a = 0.75$. In the cell \vec{n} the reaction is chosen by a linear selection method, because the number of possibilities usually is small $O(1) \dots O(10)$.

3. Selecting a transition i within cell \vec{n} ,

1. random number $r \leftarrow \text{rnd}$, sum $s \leftarrow 0$,
2. for all diffusion-transitions j :
 - (a) sum $s \leftarrow s + Q_{\vec{n}}^j$,
 - (b) if $s \geq rQ_{\vec{n}}$ then return step j ,
else next j ,
3. for all chemical reactions i ,
 - (a) sum $s \leftarrow s + Q_{\vec{n}}^i$,
 - (b) if $s \geq rQ_{\vec{n}}$ then return step i ,
else next i

The changes in all cells and classes involved have to be registered by bookkeeping steps which require the computation of the reaction-rates $Q_{\vec{n}}^i$, the local reactivity of a cell $Q_{\vec{n}}$, its class reactivity ℓ_z and the global reactivity Q . For a diffusion-step concerning a further cell \vec{m} the bookkeeping has to be done for these cells, too.

4. Bookkeeping for all cells \vec{n}, \vec{m} involved

For all cells \vec{n} , as far as their rates are concerned do

1. compute the local rates $Q_{\vec{n}}^i$,
2. compute the new cell reactivity $Q'_{\vec{n}}$ and its new class
 $z' \leftarrow \text{floor}(\text{ld}(Q'_{\vec{n}}))$,
3. if the class has not changed, i.e. $z = z'$,
then update $\ell_z \leftarrow \ell_z - Q_{\vec{n}} + Q'_{\vec{n}}$,
else delete cell \vec{n} in class z and insert it into z' ,
including an update $\ell_z \leftarrow \ell_z - Q_{\vec{n}}$ and $\ell_{z'} \leftarrow \ell_{z'} + Q'_{\vec{n}}$.
4. update the total reactivity $Q \leftarrow Q - Q_{\vec{n}} + Q'_{\vec{n}}$ and the cell reactivity $Q_{\vec{n}} \leftarrow Q'_{\vec{n}}$,

At this position the algorithm has been presented except for the data structure. Inserting and deleting of cells is somewhat delicate and will be the subject of the implementation section.

4.2.2 Implementation of the event handler

Our approach makes use of a lot of sophisticated, non-standard and non-numeric algorithmic structures. Because we do not see any way to obtain these results by usual programming methods, we want to introduce our data structures in details. The procedures dealing with these structures have been presented in the previous section.

Our program has been developed in C to achieve the highest portability and speed. We have written the code in an object oriented style nevertheless using only the ANSI C 2.0 standard. We implemented the following duties

1. an initialisation procedure, which serves as a constructor, to define the numbers of events and classes to be managed,
2. a procedure, which can be seen as a destructor, to clean up the data structures,
3. a random generator drawing a cell according its reactivity due to its contribution following the algorithm of the section above,
4. a subroutine to insert an event,
5. another subroutine for the deletion of an event.

4.2.3 Basic data structures

The most flexible data structures are pointers, thus an event is described by a pointer to the cell being represented. Because we do not want to make any assumption about the cell itself, we have implemented an event as pointer to the empty type, i.e. the type `event` is equivalent to the type `(void *)`. This choice has the advantage that it may easily be converted to a more problem-adapted structure. The event selection has been kept completely separated to encapsulate the problem-independent structure of the event handler from the problem-related structure of the cells and the topology.

The `cell`-structure contains all informations about the contents of the cell, in our case the integer numbers of A or A and B molecules. Unfortunately we found that the number of events contained in each class changes rapidly in unpredictable way. It is not possible to specify an a priori strategy to determine the size of a certain class in advance. We have therefore been forced to write an reorganisation procedure balancing the memory by dividing it into parts proportional to the size of each class.

The link `event` \rightarrow `cell` is used for the selection mechanism, i.e for events caused by a reaction of \vec{n} . This may be a chemical reaction within \vec{n} or a diffusion-step originating in \vec{n} .

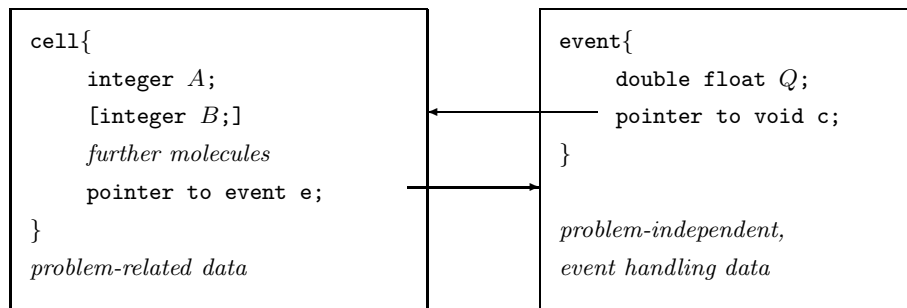


Figure 2: The cell-event data structure

The link from `cell` \rightarrow `event` is necessary to respond to changes in cell \vec{n} triggered by other cells \vec{m} concerning \vec{n} , i.e. some molecule diffusing into \vec{n} .

Since the `event`-structure may be moved by memory reorganisations, the `cell` \rightarrow `event` link has to be updated every time an event is moved. This is achieved by addressing via the link toward the other direction `event.cell.e` \leftarrow `event`. In general, the structure moving has to inform its related partner about its new address.

The cells and the events are stored in two separate arrays. The cells are arranged in a one dimensional array of size L^d using helical boundary conditions. The topological neighbourhood of a single cell belonging to a certain index array $\mathbf{n} = \sum_{\delta=0}^{d-1} L^\delta n_\delta$ with $0 \leq \mathbf{n} \leq L^d - 1$ can be addressed by

$$\mathbf{n} \pm L^\delta \text{ modulo } L^d, \quad 0 \leq \delta \leq d-1. \quad (54)$$

This way the topology has been implemented independent of the integer dimension d . For chemical reactions in fractal dimensions a more sophisticated structure is required. In this case the neighbourhood of a cell has a much more complicated topology.

The memory overhead caused by the double link between event and cell data structures is small and becomes less important with an increasing number of reacting species [7].

4.2.4 Data structure of the logarithmic classes

The array of events has a substructure appropriate to the contents of the logarithmic classes. For each class a descriptor contains all related information.

```

class{
    pointer to next class  $L_{z_{i+1}}$ ;  sequence of classes as a linked list
    integer  $z_i$ ;                    the power of 2 related to the class
    derived information             for speedup only
    double float  $\ell_z$ ;              class reactivity
    integer  $\nu_z$ ;                   number of events in class
     $F_z[]$  as a pointer to  $e$ ;        the first element of the class
}

```

Within each class the events are stored in a one dimensional array, which is to contain no empty places. The classes are realized as structures containing several elements, as shown above. The

sequence for the linear selection is realized by a linked list, so there is a pointer to the next class. The integer z contains the power of two managed by the structure. For speedup, some related frequently used information as $2^z, 2^{z+1}$ etc. also may be stored in this structure. The number ν_z describes the variable size of the array of events $F_z[]$, which is implemented as a pointer to its first element. At this point, we use the ability of the programming language C to access entries by indirect or relative addressing. The complete data structure including the links to the cells is shown by figure 3.

If an event is inserted, it is simply stored at the end of the (sub)array $F_z[]$, after that $\nu_z \leftarrow \nu_z + 1$ is increased. If an event i is deleted, gaps in the array of events $F_z[]$ must be avoided to keep the von Neumann rejection efficient. Therefore the last event of the sub-array has to be moved in the gap, $F_z[i] \leftarrow F_z[\nu_z]$, before decreasing $\nu_z \leftarrow \nu_z - 1$. Because there is a cell structure pointing to this event, its cell has to be informed that its related event has moved.

Since there is no a-priori strategy to estimate the strong fluctuations in the size of the classes, we run into the problems of collisions in memory. If the array of events $F_z[]$ of a single class L_z is going to overwrite the array of events $F_{z+1}[]$ of its successor, it is necessary to reorganise the division of the memory occupied by the array of events. This is done by crunching and re-expanding all classes proportional to their actual memory consumption.

4.3 Implementation of the reactions studied

According to our approach to separate the event handling from the chemistry within a cell, most of the work is done once the event handler has been implemented. The cells are realized as structured variables according to section (4.2.4). Thus adapting the algorithm to any specific RDS does not require more than the implementation for drawing a reaction within a cell to select the specific reaction. In our case we have to choose

1. a volume Ω ,
2. the number of cells L^d defining a discretisation $\omega = \Omega/L^d$ and $h = \omega^{1/d}$,
3. a hopping rate $\lambda_{\alpha\vec{n}\vec{m}} = D_\alpha/h^2$,
4. and a local reaction-constant $\lambda_k = \omega k/\omega^2 = k/\omega$.

Another procedure has to be provided to do the related changes and to compute the reaction-rates. According to (16) and (35) the reaction- and diffusion-rates of cell \vec{n} are given by

$$\Lambda_{\vec{n}} = \lambda_k X_{A\vec{n}} (X_{A\vec{n}} - 1) \quad \text{or} \quad (55)$$

$$\Lambda_{\vec{n}} = \lambda_k X_{A\vec{n}} X_{B\vec{n}} \quad \text{and} \quad (56)$$

$$\Lambda_{\alpha\vec{n}} = X_{\alpha\vec{n}} \lambda_{\alpha\vec{n}\vec{m}} \quad \text{with } \alpha = A, B. \quad (57)$$

5 Simulations in several dimensions

The aim of our simulations is to test the accuracy of the scaling ansatz. We want to examine both limits of the reaction-diffusion-system. On the one hand, there is the reaction-controlled limit, where the dynamics is determined by the reaction-constant $D/\Omega^{2/d} \gg k$, expecting mean-field behaviour $n_\alpha(t) \propto t^{-1}$, on the other hand we examine diffusion-controlled limit $D/\Omega^{2/d} \ll k$, where the long time annihilation is delayed by the necessity of transport.

Defining length and time scales we have chosen $\Omega = 1$ and $D = D_A = D_B = 1$. This way the reaction constant k is fixed, too. Since k now has the meaning of a rate in the case of one molecule per unit volume, D and k get inverse times $[D] = [k] = [1/t]$.

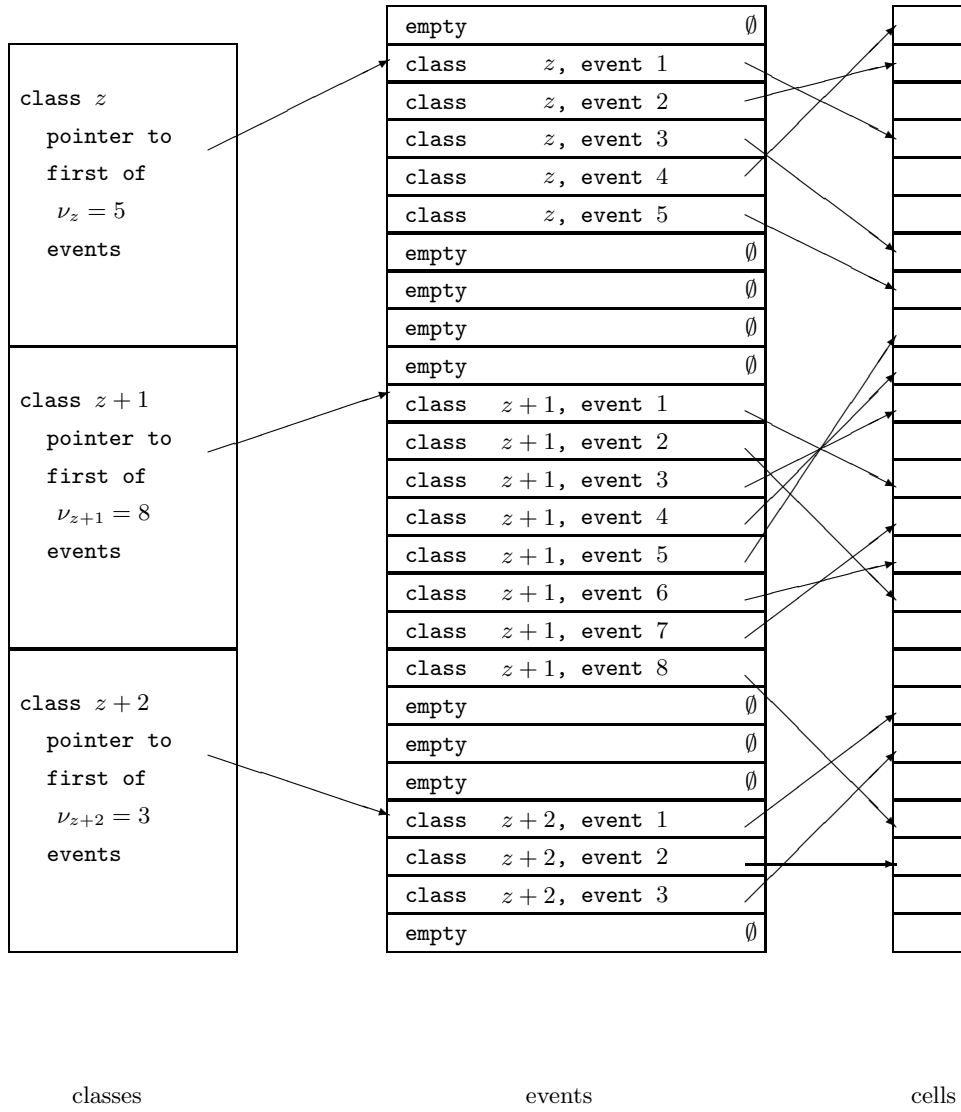


Figure 3: Data structure of classes, events and cells. At first a class is selected, then an event related to a cell, and finally a reaction in a cell. The class structure points to the first of its events. Among the event sub-arrays of different classes empty places are left.

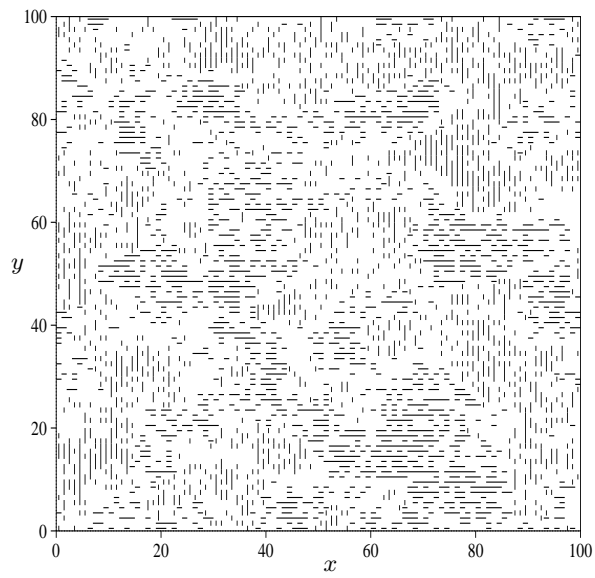


Figure 4: Formation of domains in the diffusion-controlled reaction $A + B \rightarrow \emptyset$ for a volume of 100×100 cells. The symbol ”|” denotes cells containing one or more A molecules while the symbol ”—” denotes B .

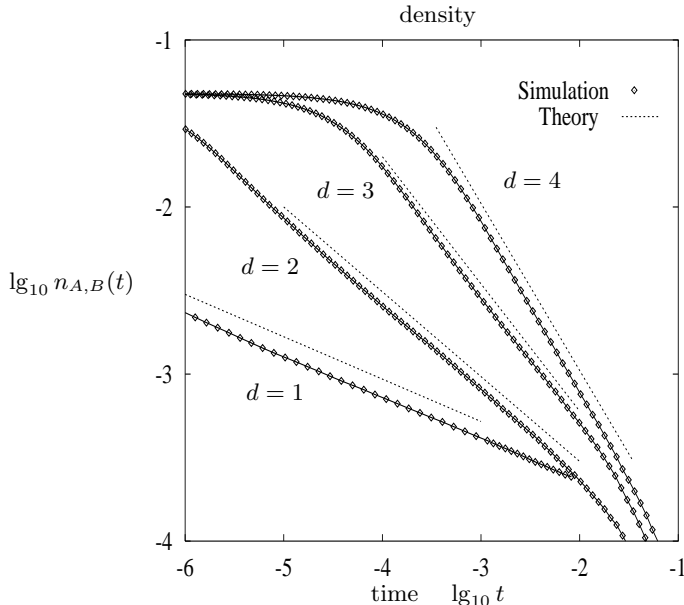


Figure 5: Simulation of $A + B \rightarrow \emptyset$ for diffusion-controlled reactions in the dimensions $d = 1, 2, 3, 4$, extension $L^d = 3 \cdot 10^4, 400^2, 50^3, 25^4$, $\Delta n = 0$

5.1 The system $A + B \rightarrow 0$

Figure (4) shows a snapshot of the formation of domains, figure (5) shows the time dependent density $n_\alpha(t)$ for the diffusion-controlled $A + B \rightarrow \emptyset$ reaction. The logarithmic plot compares the results of our simulations to straight lines $t^{-d/4}$ predicted by theory. The size of the system obviously is sufficiently large, thus one run in each dimension may be considered as self-averaging. The figures (6) and (7) show the exponents of the algebraic annihilation for both diffusion- ($D \ll k$) and reaction- ($D \gg k$) controlled limits. In our simulations these inequalities are realized by a ratio of at least 9 orders of magnitude for $k \gg D = 1$. To be sure to be sufficiently close to the limit $k/D \rightarrow \infty$ the rate k has been increased until the exponents measured by linear regression have not changed any more.

Obviously, the scaling theory for the $A + B \rightarrow \emptyset$ reaction is within the statistical errors of a few percent.

5.2 The system $A + A \rightarrow 0$

For the $A + A \rightarrow \emptyset$ reaction we observe logarithmic behaviour in its critical dimension, as predicted by [17]. This is not a multiparticle effect and results from two dimensional Smoluchowsky theory for only two particles. The $\frac{\log(t)}{t}$ has divided by $\log(t)$ to reproduce the correct t^{-1} scaling. The results have been obtained by $k = 10^8 = 10^9 \dots 10^{12}$, thus we are convinced, that our simulations give an excellent approximation to the diffusion-controlled limit. The simulations have been done with some 10^7 molecules. For every case studied, we have performed a finite size control by doubling the size L until we have not find any influence of the finite volume. Furthermore, we could neglect corrections, even if in $d = 4$ where we have been

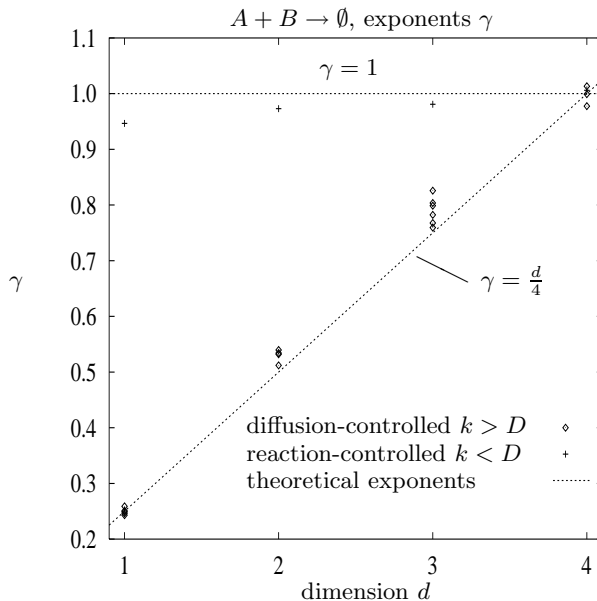


Figure 6: Simulation of $A+B \rightarrow \emptyset$ for diffusion-controlled reactions in the dimensions $d = 1, 2, 3, 4$, numerical values in table (1)

limited to a hypercube of linear extension $L = 25$, what may be a consequence of $d_c = 4$.

6 Conclusion

Thus within statistical errors of $1 \dots 2\%$ and an uncertainty due to the method of linear regression of the same order of magnitude, the tables (1) and (2) show, that scaling theory describes the $A+A \rightarrow \emptyset$ and $A+B \rightarrow \emptyset$ correctly, including the important cases $A+A \rightarrow \emptyset$ and the logarithmic corrections in $d = 2$, where we found an annihilation according to $n_A(t) \propto \frac{\log t}{t}$.

Thus, we have shown that for the simplest reaction-diffusion-systems, in which the influence of the finite number of molecules in a single cell must be taken into account, the algorithm is able to cover the whole field from diffusion- to reaction-controlled systems. A detailed study of the crossover of both regimes which determines the crossover time depending on the size of the system and the diffusion constants will follow. For the simplest reactions, the theoretical results have been reproduced nearly exactly.

Our approach follows a strategy opposite to the common cellular automaton philosophy. Instead of simulating a lot of simple diffusion-steps represented by integer operations, we try to find a length scale below which diffusion may be neglected. On this scale one single hopping process may replace a large number of infinitesimal automaton lattice gas steps.

The price to pay is the implementation of a complex event handler to maintain the exact Markoff-probabilities. From our experience we are pleased that the total overhead produced by the event handler is less than a factor of 3-4 per Markoff-event compared to reaction-systems without diffusion. The requirement of diffusion being much faster than chemical reactions

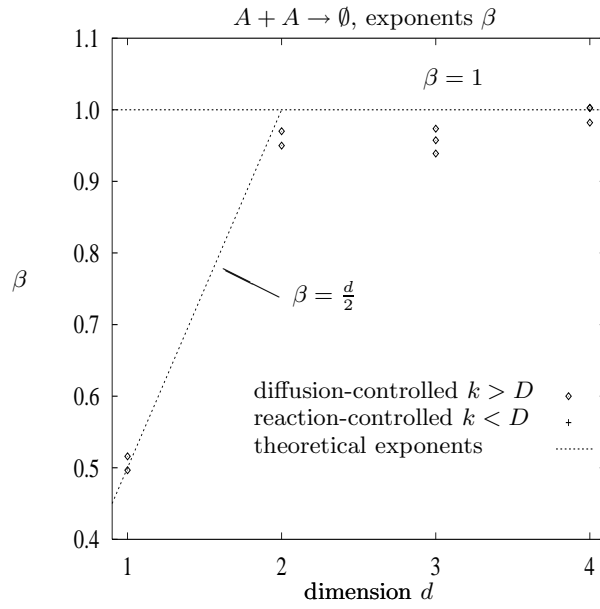


Figure 7: Exponents of the algebraic decay of $A + A \rightarrow \emptyset$ for different dimensions and several reaction-probabilities $k = 10^s$, numerical values in table (2)

d	1	2	3	4
γ_s	0.250 ₉	0.534 ₉	0.826 ₄	0.978 ₆
	0.256 ₉	0.512 ₉	0.803 ₅	0.999 ₇
			0.799 ₅	
			0.782 ₆	
γ_∞	0.249 ₁₂	0.532 ₁₁	0.768 ₉	1.013 ₉
	0.243 ₁₂	0.540 ₁₁	0.759 ₉	
	0.246 ₁₂			

Table 1: Numerical results of $A + B \rightarrow \emptyset$, see figure (6). The values are presented as γ_s , where the subscript s denotes the order of magnitude of $k = 10^s$ with γ_∞ representing the best approach to the diffusion-controlled limit, extension $L^d = 30\,000^1, 400^2, 50^3, 25^4$, $n_A(0) = n_B(0)$

d	1	2	3	4
β_s	0.5160 ₁₁			0.9821 ₇
β_∞	0.4966 ₁₂	0.9686 ₉	0.9390 ₉	1.0028 ₉
		0.9504 ₉	0.9736 ₉	1.0025 ₉
			0.9575 ₉	

Table 2: Numerical results of $A + A \rightarrow \emptyset$, see figure (7). As in the previous table the subscript s denotes the order of magnitude of the reaction constant. Note, that in $d = d_c = 2$ logarithmic corrections have been performed.

causes an additional overhead by a factor of 5–10.

The typical speed of a common Ultrix or Alpha 80 Mc RISC workstations leads to some $10^4 - 10^5$ Markoff-operations per second. For some 10^7 annihilation processes we need some 10 minutes to a few hours.

We claim our algorithm to be superior to any lattice gas simulation, because we are not restricted to one molecule per cell. For $k \rightarrow \infty$ the lattice gas is included as limit, for $k \rightarrow 0$ the simulation degenerates to the homogeneous case. An upper limit for the size ω of the reaction cells is given by the comparison of reaction and diffusion-time-scales.

The computing time for more complex RDS will increase linearly with the number of reactions. In our simple cases the memory overhead also is about a factor of 2, and can be compared to two additional components of the RDS. However, after we have implemented an event handler which is completely independent of physical problem and the underlying topology, further simulations of RDS are realized with little effort. The event handler can be easily integrated into any program, because the only necessary adaptations are the selection of the reactions in a single cell.

6.1 Future extensions

Due to the generality of the method, we see a lot of possible applications.

6.1.1 Obvious generalisations

Further extensions are possible. If its neighbourhood influences the reactions of a cell, i.e. if we want to describe surface dynamics or the fluctuations of a field, the general method keeps

unchanged. Only a slight modification of the update of the neighbour cells must be taken into account.

The simplest assumption, the chemical reactions being independent of \vec{n} , may be completed by reactions depending on the single cell or a cell and its topological neighbourhood. With regard to these conditions we only have to supplement further indices if necessary. The transition probabilities, which do not only depend on the single cell \vec{n} , but also on its neighbourhood may be denoted by the symbol $M_{\vec{n}\vec{m}\vec{m}'\dots}$. This way we can indicate the possibility that a step depends on the contents of more than one single cell. For those types of events the symbol M is in contrast to Λ , which shall be left to local rates. Therefore $M_{\vec{n}\vec{m}\vec{m}'\dots}^i$ depends on the cells $\vec{n}, \vec{m}, \vec{m}' \dots$, whereas $\Lambda_{\vec{n}}^i$ is restricted to \vec{n} . If we provide $\Lambda_{\vec{n}\vec{m}\vec{m}'\dots}^i$ with additional indices, this symbol serves as a denotation for a transition changing the contents of several cells $\vec{n}, \vec{m}, \vec{m}'$ with a rate depending only on the single cell \vec{n} . This is the manner we have described diffusion in Gillespie's algorithm by a reaction changing the contents of two cells, whose rate $\Lambda_{\alpha\vec{n}\vec{m}}$ only depends on \vec{n} alone.

To handle a reaction as a Markoff-event by the event handler, it is always assigned to its first cell specified. Thus $\Lambda_{\vec{m}\dots}^i$ and $M_{\vec{n}\vec{m}\dots}^i$ both are assigned to cell \vec{n} .

$$\begin{aligned} Q_{\vec{n}} &= \Lambda_{\vec{n}} + M_{\vec{n}} \\ &= \sum_i \Lambda_{\vec{n}}^i + \sum_i \Lambda_{\vec{n}\vec{m}}^i + \dots + \sum_i M_{\vec{n}}^i + \sum_i M_{\vec{n}\vec{m}}^i + \sum_i M_{\vec{n}\vec{m}\vec{m}'}^i + \dots \end{aligned} \quad (58)$$

The difference of the dependencies on one or more cells is unimportant for the principal selection of a reaction step. Again an arbitrary event is unambiguously denoted by i and \vec{n} and its rate $Q_{\vec{n}}^i$. More precisely, this symbol now unifies the description of $\Lambda_{\vec{n}}^i, \Lambda_{\vec{n}\vec{m}}^i, M_{\vec{n}\vec{m}\dots}^i$. For a transition touching further cells \vec{m}, \vec{m}' the bookkeeping had to be done for these cells, too.

For RDS the case of a reaction depending on several cells does not attend, we have presented this generalised formalism to show, that even more complex dynamics could be handled. This way it is possible to introduce events depending on gradients into the framework of chemical reactions. This opens an outlook not only to the simulation of surface dynamics, but points out to new approach to field theoretical models on computers.

We want to proceed to simulate more complex behaviour, with a variety of possible chemical reactions.

6.1.2 Automatically generated code

Therefore, we are writing a code generator, generating C source code by a script to automatise the process from the problem definition to the program running. As we have seen, more complex reactions are difficult to implement because for any practical purpose, any efficient code expands from one line definition to one page C source code. This will be the subject of a future paper.

We expect even much more interesting stochastic effects for more complex systems or in more complicated topologies. Simulations become essential, because by the lack of a simple scaling ansatz we may lose control over the fluctuations in a complex system, especially if the numbers of molecules in an individual cell are small.

References

- [1] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam 1981
- [2] D. E. Knuth, *The Art of Computer Programming, Vol. 1-3*, Addison-Wesley 1973-1981

- [3] I. M. Sokolov, JETP Lett. 44 67 (1986)
- [4] I. M. Sokolov, H. Schnörer, A. Blumen Phys. Rev. A. **44** 2388 (1991)
- [5] G. Zumofen, J. Klafter, A. Blumen Phys. Rev. A. **44** 8390 (1991)
- [6] T. Fricke, J. Schnakenberg, Z. Phys. B. 83(2) 1991, pp. 277
- [7] T. Fricke, *Monte-Carlo-Simulationen von Reaktions-Diffusions-Systemen mit Anwendung auf die Biophysik der Photorezeption*, diploma thesis, RWTH Aachen 1989
- [8] W. Toussaint, F. Wilczek, J. Chem. Phys. **78** 2642 (1983)
- [9] K. Kang, S. Redner, Phys. Rev. Lett. **52** 955 (1984)
- [10] K. Kang, S. Redner, Phys. Rev. A **32** 435 (1985)
- [11] A. Blumen, J. Klafter, G. Zumofen, in *Optical Spectroscopy of Glasses*, ed. Zschocke, Riedel, Dordrecht, (1986)
- [12] D. T. Gillespie, J. of Comp. Phys., 22, 1976, pp. 403–434
- [13] D. T. Gillespie, J. of Comp. Phys., 28, 1978, pp. 435–450
- [14] S. Karlin, H. M. Taylor, *A First Course in Stochastic Processes*, Academic Press, New York 1975
- [15] S. Karlin, H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press, New York 1975
- [16] V. Kuzovkov, E. Kotomin, Rep. Progr. Physics. **51** 1479 (1988)
- [17] M. Bramson, L. Lebowitz Phys. Rev. Lett., 61 (21) 1988 pp. 2397
- [18] M. Bramson, D. Griffeath, Z. Wahrscheinlichkeitstheorie, Geb. 53, 183, 1980
- [19] P. Hanusse, H. Blanché, J. of Chem. Phys., Vol. 74, No. 11, 1 June 1981, pp. 6148
- [20] D. ben-Avraham Journal of Chemical Physics Vol. 88, No. 2, 15 January 1988, pp. 941
- [21] D. Wendt, *Simulation von Reaktions-Diffusions-Systemen in ganzzahligen und fraktalen Dimensionen*, diploma thesis, RWTH Aachen 1994

: Scaling approach for $A + A \rightarrow \emptyset$ In the diffusion-controlled limit, the mean field behaviour cannot correctly describe the role of fluctuations. The approach of the scaling theory is to derive arguments connecting fluctuations to a length scale ξ . The length ξ is related to a typical time scale t_ξ by standard diffusion-scaling $t_\xi = \xi^2/D$.

Kang and Redner have argued [9], that the initial mean distance $\xi = a_{NN}$ of a A molecule to its next neighbour is proportional to $n(0)^{-1/d}$. Therefore, the time-scale τ_{NN} to pass this distance is scaled by the standard diffusion-scaling of space and time according to

$$\tau_{NN} \cong \frac{a_{NN}^2}{D} = n(0)^{-\frac{2}{d}}. \quad (59)$$

The density is scaled by a function as $g(x)$

$$n(t) = n(0) \cdot g\left(\frac{t}{\tau_{NN}}\right). \quad (60)$$

with the asymptotic properties

$$x \ll 1 \quad g(x) = 1, \quad (61)$$

$$x \gg 1 \quad g(x) \propto x^{-\beta}. \quad (62)$$

To compute the scaling exponent β , we need the additional assumption, that the long time behaviour is independent of the initial concentration, thus we obtain

$$(Dt)^\beta n(t) = \text{constant}. \quad (63)$$

In d_c the scaling and the mean-field exponents meet, thus

$$\beta = \frac{d}{2}, \quad d_c = 2. \quad (64)$$

: Scaling approach for $A + B \rightarrow \emptyset$ The dynamics of this system are governed by domains, which are defined as local areas where either the A or the B species dominates. A typical picture obtained by simulations in $d = 2$ is shown in Fig. 4.

Because A and B annihilate as pairs, the difference

$$\Delta n = n_A(t) - n_B(t) = n_A(0) - n_B(0) = \text{constant} \quad (65)$$

is conserved. In a volume ζ^d of linear extension ζ the initial number of molecules of each species may vary due to fluctuations at time $t = 0$

$$N_\alpha(0) = \langle N_\alpha \rangle \pm \sqrt{\langle N_\alpha \rangle} = n_\alpha(0)\zeta^d \pm \sqrt{n_\alpha(0)\zeta^d}. \quad (66)$$

Some species may locally be in majority even it is global in minority i.e. $N_A > N_B$ even if $n_A < n_B$. Thus by fluctuations local inversions are possible. The largest possible volume ξ^d , in which an inversion may occur, is estimated by equating the numbers of A and B molecules

$$0 = N_A(0) - N_B(0) = |\sqrt{N_A} + \sqrt{N_B}|. \quad (67)$$

Inserting (66) we solve (67) obtaining the length scale

$$\xi = \left| \sqrt{n_A(0)} - \sqrt{n_B(0)} \right|^{-\frac{2}{d}}. \quad (68)$$

By diffusion-scaling ξ is related to the time required to cross a domain

$$t_\xi = \frac{\xi^2}{D}, \quad (69)$$

assuming both diffusion-constants being equal, i.e. $D = D_A = D_B$. The scaling ansatz

$$n_\alpha(t) = C_\alpha t^{-\gamma} f_\alpha\left(\frac{t}{t_\xi}\right) \quad (70)$$

requires the existence of scaling functions $f_\alpha(x)$, whose exponential decay dominates the algebraic decay $t^{-\gamma}$ for $t > t_\xi$. For short times the scaling functions are assumed to be constant, thus

$$x \ll 1 \quad f_\alpha(x) = 1, \quad (71)$$

$$x \gg 1 \quad f(x) \propto \exp(-cx). \quad (72)$$

Inserting f_α into (65) we get

$$C_A \cdot f_A\left(\frac{t}{t_\xi}\right) - C_B \cdot f_B\left(\frac{t}{t_\xi}\right) = \frac{1}{2} \left| \sqrt{n_A(0)} - \sqrt{n_B(0)} \right| \cdot t_\xi^{-\frac{d}{4}} t^\gamma. \quad (73)$$

The left side of the equation only depends on the ratio t/t_ξ , therefore this statement yields for the right side of this equation, too, thus leading to the algebraic exponent

$$\gamma = \frac{d}{4}, \quad d_c = 4. \quad (74)$$

Although in the limit $n_A(0) \rightarrow n_B(0)$ the factor $\left| \sqrt{n_A(0)} - \sqrt{n_B(0)} \right| \rightarrow 0$ in equation (73) disappears, the argumentation is not affected and the scaling keeps valid [9, 10].